In the Specification

Applicant presents replacement paragraphs below indicating the changes with insertions indicated by underlining and deletions indicated by strikeouts and/or double bracketing.

Please replace paragraph 2 beginning at page 2 line 12 with the amended paragraph/line as follows:

To illustrate the conventional method of cache utilization, reference is made to the database index tree 190 of prior art Fig. 1 and the flowchart of prior art Fig. 2. The index tree 190 contains a set of nodes 100-132, each of which represents an alphabetical range in a database. The nodes 100-132 contain pointer 134-180. The pointer 134-164 contained in the nodes 102-116 point to other nodes, while the bottom row of nodes 118-132 contain pointers to parts of a desired data block 195. For example, the node 100 contains pointers 134 and 136, which point to the nodes 102 and 104 respectively. Additionally, the bottom row of ~~modes~~ nodes 118-132 are horizontally linked, although the linking pointers are not depicted.

Please replace paragraph 3 beginning at page 6 line 16 with the amended paragraph/line as follows:

When a pipeline program is executed on a multiprocessor system, each CPU will seek out a server having a pending work ~~packets~~ packet. No two CPU's will be permitted to process a single server's workload unless they can do so without conflict. Thus, the general effect of this scheme will be that each CPU will tend to perform a single task for all users, thereby insuring that the global context state for each task tends to remain in the cache of the CPU performing that task. The user state information is passed from server to server within the work packets.

Please replace paragraph 2 beginning at page 7 line 3 with the amended paragraph/line as follows:

PRIOR ART Fig. 2 is a flowchart generally depicting the steps for searching [[a]] an index tree using a conventional programming technique;

Please replace paragraph 3 beginning at page 14 line 21 with the amended paragraph/line as follows:

If the server is exclusive, then the CPU searches for another server at step 600. If the server is partitioned, then the CPU reads a value of the first available work packet of the server and determines whether the value satisfies the condition for which a multiple instance of the server may be run (i.e. the partitioning condition). If the condition is not satisfied, the CPU searches for another server having processed work packets in its associated queue at step 600. If the partitioning condition is satisfied for a work packet, the flow continues at step 606. At steps 606-608, the CPU performs the operations on each work packet in the server's queue until the queue is empty. The CPU then waits for another server to receive a work packet at step 600. The steps of Fig. 6 can be rearranged and modified in many ways. For example, if the CPU finds that a work packet does not satisfy the required conditions for being processed on a partitioned server, the CPU may check other packets in the queue before attempting [[for]] to locate another server.

Please replace paragraph 2 beginning at page 15 line 11 with the amended paragraph/line as follows:

To create a pipelined program for searching the index tree 190 of Prior art Fig. 1, the data structures of Fig. 7 may be used. The exemplary node search work packet data structure 700 and I/O server work packet 702 each inherit user context information from the structure labeled "context." The user context data may include the location of a buffer to receive the results of the search, a network address to send the results to, a user ID to check access rights, and a pointer to the packet containing the original query that initiated this index tree search. A pointer 706 references the ~~parents~~ parent packet while an action code 708 ~~tell~~ tells the server whether a special function is to be performed. Such action codes may include: (1) "StartingUpSearch" to initiate an initialization procedure; (2) "StartofBatch" to cause the server to prepare for a new batch of work packets; (3) "EndofBatch" to close out processing of a batch of work packets; and (4) "ShuttingdownServer" to cause the server to exit. This list is meant to be exemplary only and there are many other specialized functions possible.

Please replace paragraph 2 beginning at page 16 line 8 with the amended paragraph/line as follows:

A set of user-defined fields for the node search work packet data structure 700 may include a pointer 712 to a value or range of values for which to search in the index tree nodes. For example, the pointer 712 might point to the search parameter "P-Q", indicating a search for nodes having pointers to data starting with letters between P and Q. Another user-defined field 714 might contain the number of the database page containing the node being searched. Finally, if the node being searched is in cache memory, a pointer 716 might contain the address of the page containing the node. In the I/O server packet definition 702, the user defined fields may include variables 718 and [[720]] 719 for holding the database page number and memory address to which to write or from which to read.